EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for the examiner's amendment was given in a telephone interview with Mr. Michael Henry, Registration No. 59,516 on July 10, 2008.

3. The claims are amended as follow:

1. (**Currently Amended**) A method for communication between a first computer operating in a first object-oriented run-time environment and a second computer operating in a second, different object-oriented run-time environment, the method comprising:

receiving a first message with an object identification and an action identification from the first computer to the second computer;

identifying an object in the second run-time environment according to the object identification;

verifying an existence of an action, according to the action identification, in the identified object in the second run-time environment;

determining an action representation of **the** ~~**an**~~ action, according to the action identification, in the second run-time environment for the identified object; and

executing the action using the action representation, wherein executing the action includes:

converting a request identification that is part of the action identification to a second representation for the second run-time environment using a look-up table; and

inserting the second representation into the second ~~**application**~~ **run-time environment**.

2. (**Canceled**)

3. (**Canceled**)

4. (**Original**) The method of claim 1 further comprising returning to the first computer a second message as a confirmation message with an object identification and a response identification.

5. (**Original**) The method of claim 4 further comprising displaying, using the first computer, at least a portion of the response identification.

6. (**Previously Presented**) The method of claim 1, wherein executing the action further comprises:

extracting a second property representation of a property identified by the action identification;

converting the second property representation to a first property representation for the first run-time environment; and

returning to the first computer a second message as a result message with an object identification and a response identification, the response identification indicating the first property representation for the first run-time environment.

7. (**Previously Presented**) The method of claim 6 further comprising displaying, using the first computer, at least a portion of the response identification.

8. (**Previously Presented**) The method of claim 1, further comprising: receiving a second message with a second object identification and a second action identification from the first computer to the second computer;

identifying a second object in the second run-time environment according to the second object identification;

determining a second action representation of a second action, according to the second action identification, in the second run-time environment for the identified second object; and

executing the second action using the second action representation, wherein executing the second action further comprises:

converting a function identification and a parameter identification of the second action identification to function and parameter representations for the second run-time environment;

performing a function that is identified by the second action identification using the function and parameter representations for the second run-time environment;

converting parameters that result from performing the function into parameter representations for the first run-time environment; and

returning a third message to the first computer with an object identification and a response identification, with the response identification indicating the parameter representations.

9. (**Original**) The method of claim 8 wherein converting parameters uses a look-up table.

10. (**Currently Amended**) A computer program product used in a communication system of a first computer with a first object-oriented run-time environment and a second computer with a second, different object-oriented run-time environment, wherein the first computer sends a first message with an object identification and an action identification to the second computer, the computer program product embodied on a carrier and having computer code instructions to cause a processor of the second computer to interpret the first message, the instructions comprising:

code for identifying an object in the second run-time environment according to the object identification;

code for verifying the existence of an action, according to the action identification, in the identified object in the second run-time environment;

code for determining a representation of **the** ~~**an**~~ action, according to the action identification, in the second run-time environment for the identified object; and

code for executing the action using the representation, wherein the code for executing includes:

code for converting a request identification that is part of the action identification to a further representation for the second run-time environment; and

code for inserting the further representation into the second ~~**application**~~ **run-time environment**.

11. (**Canceled**)

12. (**Previously Presented**) The computer program product of claim 10 wherein the instructions further comprise code for returning a second message as a confirmation message to the first computer, the second message including an object identification and a response identification.

13. (**Canceled**)

14. (**Previously Presented**) The computer program product of claim 10 wherein the code for converting uses a look-up table.

15. (**Previously Presented**) The computer program product of claim 10 wherein the code for executing comprises:

code for extracting a second property representation of a property identified by the action identification;

code for converting the second property representation to a first property representation for the first run-time environment; and

code for returning to the first computer a second message as a result message with an object identification and a response identification, the response identification indicating the first property representation for the first run-time environment.

16. (**Previously Presented**) The computer program product of claim 10, wherein the code for executing comprises:

code for converting a function identification and a parameter identification of the action identification to function and parameter representations for the second run-time environment;

code for performing a function that is identified by the action identification using the function and parameter representations for the second run-time environment;

code for converting parameters that result from performing the function into parameter representations for the first run-time environment; and

code for returning a second message to the first computer with an object identification and a response identification, with the response identification indicating the parameter representations.

17. (**Currently Amended**) A computer communication system comprising a first computer operating in a first object-oriented run-time environment and a second computer operating in a second, different object-oriented run-time environment, wherein

the first computer sends a first message with an object identification and an action identification to the second computer, the second computer comprising:

a first module to identify an object in the second run-time environment according to the object identification;

a second module to verify an existence of an action identified in the action identification in the identified object in the second run-time environment;

a third module to determine a representation of the action in the second run-time environment for the identified object; and

a fourth module to execute the action by using the representation and to return a second message as confirmation message to the first computer, the second message with object identification and response identification, wherein the fourth module is adapted to (a) convert a request identification that is part of the action identification to a further representation for the second run-time environment using a look-up table, and (b) insert the further representation into the second ~~**application**~~ **run-time environment**.

18. (**Canceled**)

19. (**Previously Presented**) The computer communication system of claim 17 wherein the fourth module is further adapted to:

extract a second property representation of a property identified by the action identification;

convert the second property representation to a first property representation for the first run-time environment; and

return to the first computer a second message as a result message with an object identification and a response identification, the response identification indicating the first property representation for the first run-time environment.

20. (**Previously Presented**) The computer communication system of claim 17 wherein the fourth module is further adapted to:

convert a function identification and a parameter identification of the action identification to function and parameter representations for the second run-time environment;

perform a function that is identified by the action identification using the function and parameter representations for the second run-time environment;

convert parameters that result from performing the function into parameter representations for the first run-time environment; and

return a second message to the first computer with an object identification and a response identification, with the response identification indicating the parameter representations.

21. (**Currently Amended**) A computer program product **embodied on a carrier and** comprising computer readable instructions operable when executed to **cause a processor to**:

receive a first message with an object identification and an action identification from a first computer with a first object-oriented run-time environment;

identify an object in a second computer with a second, different object-oriented run-time environment, according to the object identification;

verify the existence of an action, according to the action identification, in the identified object in the second run-time environment;

determine a representation of **the** ~~**an**~~ action, according to the action identification, in the second run-time environment for the identified object; and

execute the action using the representation, wherein the instructions operable to execute the action comprise instructions operable to:

convert a request identification that is part of the action identification to a further representation for the second run-time environment; and

insert the further representation into the second ~~**application**~~ **run-time environment**.

22. (**Currently Amended**) A computer program product **embodied on a carrier and** comprising computer readable instructions operable when executed to **cause a processor to**:

receive a first message with an object identification and an action identification from a first computer with a first object-oriented run-time environment;

identify an object in a second computer with a second, different object-oriented run-time environment, according to the object identification;

verify the existence of an action, according to the action identification, in the identified object in the second run-time environment;

determine a representation of **the** ~~an~~ action, according to the action identification, in the second run-time environment for the identified object; and

execute the action using the representation, wherein the instructions operable to execute the action comprise instructions operable to:

extract a second property representation of a property identified by the action identification;

convert the second property representation to a first property representation for the first run-time environment; and

return to the first computer a second message as a result message with an object identification and a response identification, the response identification indicating the first property representation for the first run-time environment.

23. (**Currently Amended**) A computer program product **embodied on a carrier and** comprising computer readable instructions operable when executed to **cause a processor to**:

receive a first message with an object identification and an action identification from a first computer with a first object-oriented run-time environment;

identify an object in a second computer with a second, different object-oriented run-time environment, according to the object identification;

verify the existence of an action, according to the action identification, in the identified object in the second run-time environment;

determine a representation of **the** ~~an~~ action, according to the action identification, in the second run-time environment for the identified object; and

execute the action using the representation, wherein the instructions operable to execute the action comprise instructions operable to:

convert a function identification and a parameter identification of the action identification to function and parameter representations for the second run-time environment;

perform a function that is identified by the action identification using the function and parameter representations for the second run-time environment;

convert parameters that result from performing the function into parameter representations for the first run-time environment; and

return a second message to the first computer with an object identification and a response identification, with the response identification indicating the parameter representations.

24. (**Previously Presented**) The method of claim 8, the second object identified in the second message being the first object identified in the first message.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Qing-Yuan Wu whose telephone number is (571)272-3776. The examiner can normally be reached on 8:30am-6:00pm Monday-Thursday and alternate Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/

Supervisory Patent Examiner, Art Unit 2195

/Qing-Yuan Wu/

Examiner, Art Unit 2194